Dołączenie systemu Linux do domeny Active Directory (realmd + SSSD)

Typ: procedura **Dotyczy: RHEL/Rocky 9–10** **Czas: ~30 min**

Efekt: system loguje użytkowników domenowych przez Kerberos, mapuje ich UID/GID przez SSSD, tworzy im katalogi domowe przy pierwszym logowaniu i pozwala wskazanej grupie domenowej używać `sudo`*.*

Skróty użyte na tej stronie (10)

Skrót	Co znaczy
AD	Active Directory: katalog Microsoftu z kontami użytkowników, komputerów i grup w firmie
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
LDAP	Lightweight Directory Access Protocol: protokół do odpytywania katalogu użytkowników (AD, FreeIPA)
LDAPS	LDAP szyfrowany TLS, zwykle port 636
PAM	Pluggable Authentication Modules: wymienne moduły logowania w Linuksie, decydują na przykład o sile wymaganego hasła
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
SMB	Server Message Block: protokół udostępniania plików i drukarek w sieci Windows, na nim stoi zwykły udział sieciowy
SRV	rekord DNS wskazujący nie adres hosta, tylko usługę i port, pod którym jej szukać, np. kontroler domeny w sieci
SSSD	System Security Services Daemon: usługa na kliencie linuksowym tłumacząca logowanie, sudo i sprawdzanie grup na zapytania do katalogu, czy to FreeIPA, czy Active Directory
UID	User ID: numer użytkownika w Linuksie

Zanim zaczniemy

> **Z ŻYCIA WZIĘTE**
>
> O 9:00 Ola dostaje nowy laptop deweloperski, który ma trafić do domeny ad.zielona.example, żeby mogła logować się swoim kontem firmowym zamiast kontem lokalnym. Tomek uruchamia `realm join`, ale polecenie wisi kilkanaście sekund i kończy się `kinit: Clock skew too great`, bo laptop stał wyłączony w szufladzie kilka tygodni i zegar poszedł w tył. Po ręcznym `chronyc makestep` dołączenie do domeny przechodzi bez dalszych przeszkód.

Wymaganie	Jak sprawdzić
DNS wskazuje na kontrolery domeny (forward i rekordy SRV)	`dig +short _ldap._tcp.NazwaDomeny SRV`
Kontroler domeny odpowiada pod pełną nazwą	`dig +short NazwaKontroleraDC`
Synchronizacja czasu — offset poniżej 5 minut	`chronyc tracking` (pole *System time*)
Porty wychodzące do kontrolera otwarte	`nc -zv NazwaKontroleraDC 88 389 445 464 636 3268`
Nazwa hosta jest pełną, rozwiązywalną nazwą (FQDN)	`hostnamectl`
Mamy konto uprawnione do dołączania komputerów do domeny	konto AD z odpowiednim delegowaniem (`NazwaUseraWDomenie`)

Porty, które musi mieć otwarte firewall między klientem a kontrolerem domeny: **88/tcp+udp** (Kerberos), **389/tcp** (LDAP), **445/tcp** (SMB, używane przez `adcli`), **464/tcp+udp** (zmiana hasła Kerberos), **636/tcp** (LDAPS) i **3268/tcp** (Global Catalog).

Uważaj: dołączenie do domeny nadpisuje `/etc/nsswitch.conf`, `/etc/sssd/sssd.conf` i stos PAM przez `authselect` — na maszynie z aktywnymi sesjami lokalnymi robimy to w oknie serwisowym, bo restart `sssd` zrywa trwające uwierzytelnianie domenowe. Dołączenie tworzy też obiekt komputera w Active Directory; gdy procedura zostanie przerwana w połowie, obiekt może zostać po stronie domeny i wymaga ręcznego usunięcia przez administratora AD.

Przebieg

1. Instalacja pakietów

```
sudo dnf install -y realmd sssd adcli samba-common-tools oddjob oddjob-mkhomedir \
  samba-common krb5-workstation openldap-clients
```

Przykład:

```
Installed:
  adcli-0.9.2-6.el9.x86_64          oddjob-0.34.7-8.el9.x86_64
  oddjob-mkhomedir-0.34.7-8.el9     realmd-0.17.1-11.el9.x86_64
  samba-common-tools-4.19.4-...     sssd-2.9.4-...
Complete!
```

`realmd sssd adcli samba-common-tools oddjob oddjob-mkhomedir` to minimalny zestaw wymagany do dołączenia i zarządzania kontem domenowym; `oddjob-mkhomedir` odpowiada konkretnie za tworzenie katalogów domowych przy pierwszym logowaniu. `krb5-workstation` i `openldap-clients` nie są niezbędne do samego dołączenia, ale przydają się do ręcznej diagnostyki (`kinit`, `ldapsearch`).

2. Rozpoznanie domeny

```
realm discover NazwaDomeny
```

Przykład:

```
ad.zielona.example
  type: kerberos
  realm-name: AD.ZIELONA.EXAMPLE
  domain-name: ad.zielona.example
  configured: no
  server-software: active-directory
  client-software: sssd
  required-package: oddjob-mkhomedir
  required-package: sssd
  required-package: adcli
  required-package: samba-common-tools
```

`realm discover` pyta DNS o rekordy SRV domeny i sprawdza, czy potrzebne pakiety są zainstalowane. `configured: no` to oczekiwany stan przed kolejnym krokiem — system jeszcze nie jest dołączony. Gdy polecenie nie zwraca nic, problem leży w DNS (patrz *Gdy coś nie zadziała*).

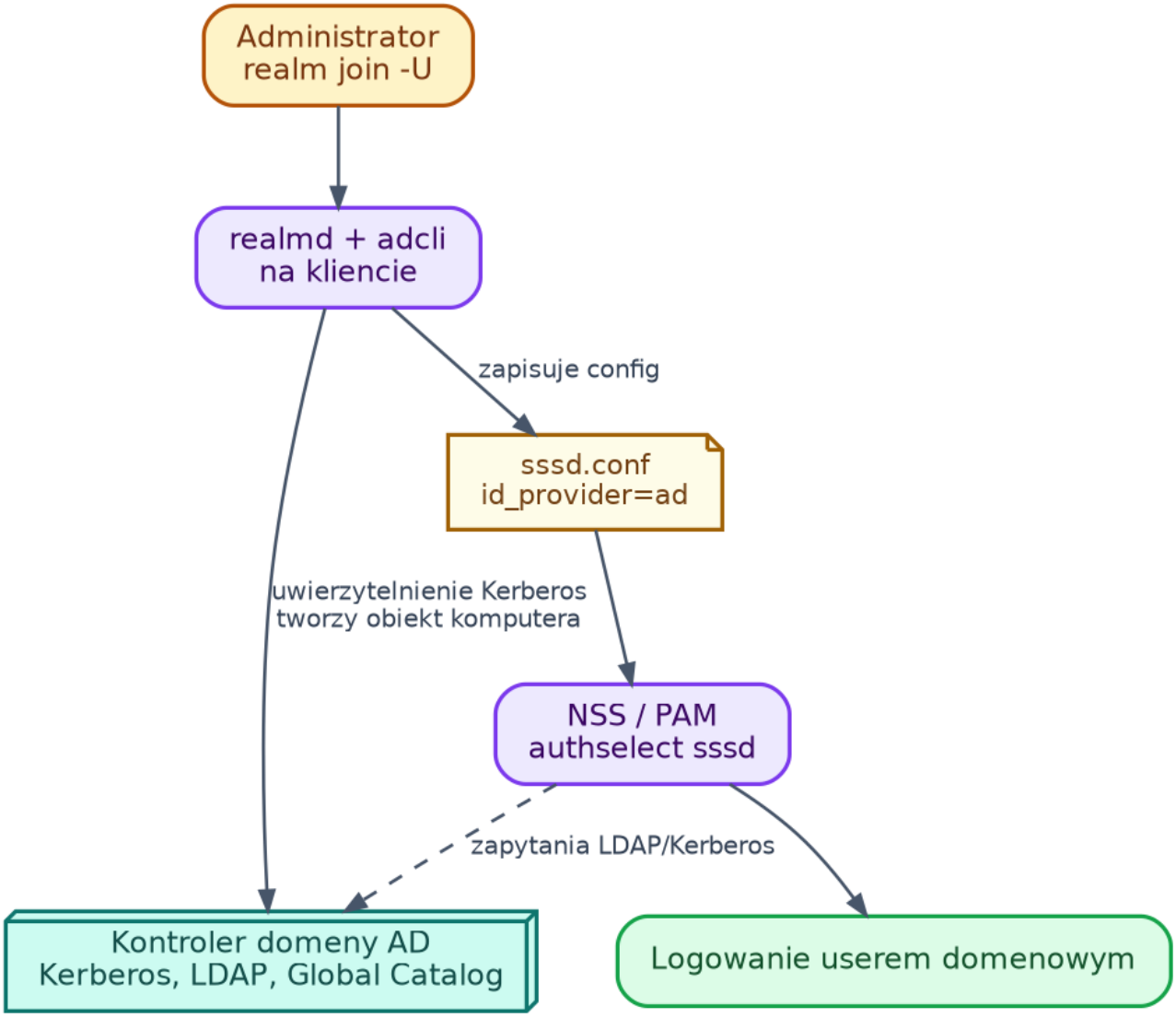
3. Dołączenie do domeny

```
sudo realm join -U NazwaUseraWDomenie NazwaDomeny
```

Przykład:

```
Password for adm_user:
```

`realm join` prosi o hasło konta wskazanego przez `-U`, wykonuje uwierzytelnienie Kerberos, tworzy obiekt komputera w AD i sam dopisuje potrzebne sekcje do `/etc/sssd/sssd.conf` oraz `/etc/nsswitch.conf`. Polecenie kończy się bez żadnego komunikatu przy powodzeniu — brak błędu to sukces.



realmd i adcli uwierzytelniają się w AD, tworzą obiekt komputera i zapisują sssd.conf, a od tej chwili NSS i PAM pytają kontroler domeny przy każdym logowaniu (kliknięcie otwiera pełny rozmiar)

4. Weryfikacja dołączenia

```
realm list
```

Przykład:

```
ad.zielona.example
  type: kerberos
  realm-name: AD.ZIELONA.EXAMPLE
  domain-name: ad.zielona.example
  configured: kerberos-member
  server-software: active-directory
  client-software: sssd
  required-package: oddjob-mkhomedir
  login-formats: %U@ad.zielona.example
  login-policy: allow-realm-logins
```

`configured: kerberos-member` potwierdza, że system jest teraz członkiem domeny. `login-policy: allow-realm-logins` oznacza, że w tej chwili **każdy** użytkownik domenowy może się zalogować — krok 8 to zawęzi.

5. Konfiguracja SSSD dla spójnego logowania

```
sudo vim /etc/sssd/sssd.conf
```

Weryfikujemy, czy sekcje dopisane automatycznie przez `realm join` plus nasze ręczne poprawki dają w efekcie taki plik:

```
[sssd]
domains = NazwaDomeny
config_file_version = 2
services = nss, pam

[domain/NazwaDomeny]
ad_domain = NazwaDomeny
krb5_realm = NAZWADOMENY
realmd_tags = manages-system joined-with-adcli
id_provider = ad
access_provider = ad
ldap_id_mapping = True
```

```
ad_hostname = NazwaKontroleraDC
dyndns_update = true
dyndns_refresh_interval = 43200
dns_update_ptr = true
dyndns_ttl = 3600
dyndns_auth = GSS-TSIG

use_fully_qualified_names = True
fallback_homedir = /home/%u@%d
override_shell = /bin/bash

krb5_store_password_if_offline = True
cache_credentials = True
```

> (gdzie `NazwaDomeny` to pełna nazwa domeny, np. `ad.zielona.example`, a `NAZWADOMENY` to ta sama nazwa wielkimi literami — tak Kerberos zapisuje realm)

`use_fully_qualified_names` wymusza logowanie jako `user@NazwaDomeny`, żeby nie kolidować z kontami lokalnymi o tych samych nazwach. `fallback_homedir` ustala wzorzec katalogu domowego zgodny z tym formatem. `override_shell` wymusza jedną powłokę logowania dla wszystkich userów domenowych, niezależnie od tego, co ma ustawione w atrybucie AD — przydaje się, gdy w AD nikt nigdy nie wypełnił pola powłoki i domyślnie wychodziłby `/bin/false`.

Plik ma prawa `600` i właściciela `root` — jeśli `realm join` je zmieni, przywracamy je przed restartem usługi:

```
sudo chmod 600 /etc/sssd/sssd.conf
```

6. Aktywacja tworzenia katalogów domowych

```
sudo authselect select sssd with-mkhomedir --force
```

Przykład:

```
Profile "sssd" was selected.
The following nsswitch maps are overwritten by the profile:
```

```
- passwd
- group
```

`authselect` podmienia konfigurację PAM i NSS na profil `sssd`, a opcja `with-mkhomedir` dopisuje moduł `pam_oddjob_mkhomedir`, który tworzy katalog domowy przy pierwszym udanym logowaniu. Bez tego kroku logowanie się powiedzie, ale user wyląduje bez katalogu domowego.

7. Restart SSSD

```
sudo systemctl restart sssd
sudo systemctl enable sssd
```

Restart wczytuje nową konfigurację; `enable` gwarantuje, że usługa wstanie po reboocie. Na tym etapie logowanie kontem domenowym już powinno działać, ale jeszcze bez sudo.

8. Nadanie sudo grupie domenowej

```
sudo realm permit -g 'GrupaWdomenie@NazwaDomeny'
echo '%GrupaWdomenie@NazwaDomeny ALL=(ALL) NOPASSWD:ALL' | sudo tee /etc/sudoers.d/ad-linux-admins
sudo chmod 440 /etc/sudoers.d/ad-linux-admins
sudo visudo -c
```

Przykład:

```
/etc/sudoers.d/ad-linux-admins: parsed OK
```

`realm permit -g` zawęża `login-policy` z kroku 4 do konkretnej grupy — od tej chwili logować się mogą tylko jej członkowie, a nie cała domena. Osobny plik w `/etc/sudoers.d/` zamiast edycji `/etc/sudoers` przez `visudo` bezpośrednio ułatwia wycofanie zmiany i nie miesza się z lokalną konfiguracją sudo. `visudo -c` sprawdza składnię przed użyciem — literówka w tym pliku blokuje **każde** wywołanie `sudo` w systemie.

9. Test logowania

```
id NazwaUseraWDomenie@NazwaDomeny
```

Przykład:

```
uid=1458001107(adm_user@ad.zielona.example) gid=1458000513(domain users@ad.zielona.example)
groups=1458000513(domain users@ad.zielona.example),1458001205(linux-admins@ad.zielona.example)
```

```
su - NazwaUseraWDomenie@NazwaDomeny
```

Przykład:

```
Creating home directory for adm_user@ad.zielona.example.
[adm_user@srv01 ~]$
```

`id` potwierdza, że SSSD widzi konto i poprawnie mapuje jego grupy — obecność grupy domenowej na liście grup oznacza, że krok 8 zadziała. Komunikat „Creating home directory" przy pierwszym `su`/logowaniu to dowód, że `oddjob-mkhomedir` z kroku 6 działa.

Weryfikacja

```
id NazwaUseraWDomenie@NazwaDomeny && sudo -l -U NazwaUseraWDomenie@NazwaDomeny 2>/dev/null;
getent passwd NazwaUseraWDomenie@NazwaDomeny
```

Przykład:

```
uid=1458001107(adm_user@ad.zielona.example) gid=1458000513(domain users@ad.zielona.example)
groups=...
adm_user@ad.zielona.example:*:1458001107:1458000513:adm_user:/home/adm_user@ad.zielona.example
:/bin/bash
```

`getent passwd`, czytający pełny wpis (nie tylko `id`), potwierdza, że NSS poprawnie sięga do SSSD, a nie tylko PAM podczas logowania. Katalog domowy z kroku 9 (`ls -ld /home/NazwaUseraWDomenie@NazwaDomeny`) powinien istnieć i należeć do tego użytkownika.

Wycofanie zmian

```
sudo realm leave NazwaDomeny
sudo rm -f /etc/sudoers.d/ad-linux-admins
sudo authselect select sssd --force
sudo systemctl restart sssd
```

`realm leave` usuwa system z domeny lokalnie i, jeśli konto użyte do wykonania polecenia ma odpowiednie uprawnienia, kasuje też obiekt komputera po stronie AD — w przeciwnym razie zostaje on osierocony i administrator domeny musi go usunąć ręcznie z konsoli AD. Sekcje domeny w `/etc/sssd/sssd.conf` zostają w pliku po `realm leave`; jeśli maszyna ma już nigdy nie wracać do domeny, usuwamy je ręcznie i przywracamy `authselect select sssd` bez `with-mkhomedir` (albo `authselect select minimal`, jeśli w ogóle nie ma być SSSD).

Gdy coś nie zadziała

Objaw	Przyczyna	Naprawa
`realm discover` nie zwraca nic albo `realm: Couldn't find hostname`	Brak rekordów SRV domeny w DNS albo klient pyta zły serwer DNS	`dig +short _ldap._tcp.NazwaDomeny SRV`; poprawiamy `/etc/resolv.conf` albo konfigurację DHCP, żeby wskazywała na DNS kontrolerów domeny
`realm join` kończy się `kinit: Clock skew too great`	Rozjazd czasu między klientem a kontrolerem domeny powyżej 5 minut (limit Kerberosa)	`chronyc tracking`; jeśli offset duży, `sudo chronyc makestep` albo poprawiamy `/etc/chrony.conf` i czekamy na synchronizację
`realm join` kończy się `Necessary packages are not installed`	Brakuje jednego z pakietów wymaganych przez `realm discover` (najczęściej `oddjob-mkhomedir` albo `adcli`)	Instalujemy brakujący pakiet z komunikatu błędu, powtarzamy `realm join`
Logowanie działa, ale użytkownik ląduje bez katalogu domowego / `No directory, logging in with HOME=/`	`authselect` nie ma aktywnego profilu `with-mkhomedir`, albo `oddjobd` nie działa	`authselect current` sprawdza aktywny profil; `sudo systemctl status oddjobd`; powtarzamy krok 6 z `--force`

Powiązane: Uwierzytelnianie PAM i authselect · SSSD: mapowanie ID i cache poświadczeń · sudo i /etc/sudoers.d

Źródła: `man 8 realm` · `man 5 sssd.conf` · `man 8 authselect` · Red Hat Enterprise Linux — *Integrating RHEL Systems Directly with Windows Active Directory*

Revision #20
Created 2025-05-05 12:43:49 UTC by Tomasz Konieczny
Updated 2026-10-03 19:09:23 UTC by Tomasz Konieczny